Oblivious Chase Termination

Oblivious Chase Termination: The Sticky Case

Marco Calautti
School of Informatics, University of Edinburgh, UK
mcalautt@inf.ed.ac.uk

Andreas Pieris
School of Informatics, University of Edinburgh, UK
apieris@inf.ed.ac.uk

Abstract

The chase procedure is one of the most fundamental algorithmic tools in database theory. A key algorithmic task is uniform chase termination, i.e., given a set of tuple-generating dependencies (tgds), is it the case that the chase under this set of tgds terminates, for every input database? In view of the fact that this problem is undecidable, no matter which version of the chase we consider, it is natural to ask whether well-behaved classes of tgds, introduced in different contexts such as ontological reasoning, make our problem decidable. In this work, we consider a prominent decidability paradigm for tgds, called stickiness. We show that for sticky sets of tgds, uniform chase termination is decidable if we focus on the (semi-)oblivious chase, and we pinpoint its exact complexity: PSpace-complete in general, and NLogSpace-complete for predicates of bounded arity. These complexity results are obtained via graph-based syntactic characterizations of chase termination that are of independent interest.

2012 ACM Subject Classification Theory of computation → Database constraints theory; Theory of computation → Logic and databases

Keywords and phrases Chase procedure, tuple-generating dependencies, stickiness, termination, computational complexity

Digital Object Identifier 10.4230/LIPIcs.ICDT.2019.17

Funding Calautti and Pieris are funded by the EPSRC grant EP/S003800/1 "EQUID", and the EPSRC programme grant EP/M025268/ "VADA".

1 Introduction

The *chase procedure* (or simply chase) is a fundamental algorithmic tool that has been successfully applied to several database problems such as containment of queries under constraints [1], checking logical implication of constraints [3, 17], computing data exchange solutions [10], and query answering under constraints [5], to name a few. The chase procedure accepts as an input a database D and a set Σ of constraints and, if it terminates, its result is a finite instance D_Σ that is a *universal* model of D and Σ, i.e., is a model that can be homomorphically embedded into every other model of D and Σ. In other words, D_Σ acts as a representative of all the other models of D and Σ. This is the reason for the ubiquity of the chase in database theory, as discussed in [8]. Indeed, many key database problems can be solved by simply exhibiting a universal model.

A prominent class of constraints that can be naturally treated by the chase procedure is the class of *tuple-generating dependencies (tgds)*, i.e., sentences of the form

$$\forall \bar{x} \forall \bar{y} \, (\phi(\bar{x}, \bar{y}) \ \rightarrow \ \exists \bar{z} \, \psi(\bar{x}, \bar{z})) \, ,$$

where ϕ and ψ are conjunctions of atoms. Given a database D and a set Σ of tgds, the chase adds new atoms to D (possibly involving null values that act as witnesses for the existentially quantified variables) until the final result satisfies Σ. For example, given the database $D = \{R(c)\}$, and the tgd $\forall x(R(x) \rightarrow \exists y \, P(x, y) \wedge R(y))$, the database atom *triggers*

the tgd, and the chase will add in D the atoms $P(c, \perp_1)$ and $R(\perp_1)$ in order to satisfy it, where $\perp_1$ is a (labeled) null representing some unknown value. However, the new atom $R(\perp_1)$ triggers again the tgd, and the chase is forced to add the atoms $P(\perp_1, \perp_2), R(\perp_2)$, where $\perp_2$ is a new null. The result of the chase is the instance

$$\{R(c), P(c, \perp_1)\} \cup \bigcup_{i>0} \{R(\perp_i), P(\perp_i, \perp_{i+1})\},$$

where $\perp_1, \perp_2, \ldots$ are nulls.

The above example shows that the chase procedure may not terminate, even for very simple databases and sets of tgds. This fact motivated a long line of research on identifying subclasses of tgds that ensure the termination of the chase procedure, no matter how the input database looks like. A prime example is the class of *weakly-acyclic* sets of tgds [10], which is the standard language for data exchange purposes, and guarantees the termination of the semi-oblivious and restricted (a.k.a. standard) chase. A similar formalism, called *constraints with stratified-witness*, has been proposed in [9]. Inspired by weak-acyclicity, the notion of *rich-acyclicity* has been proposed in [16], which guarantees the termination of the oblivious chase. Many other sufficient conditions for chase termination can be found in the literature; see, e.g., [8, 9, 13, 15, 18, 19] – this list is by no means exhaustive, and we refer the reader to [14] for a comprehensive survey.

With so much effort spent on identifying sufficient conditions for the termination of the chase procedure, the question that immediately comes up is whether a sufficient condition that is also *necessary* exists. In other words, given a set Σ of tgds, is it possible to decide whether, for every database D, the chase on D and Σ terminates? This question has been addressed in [11], and has been shown that the answer is negative, no matter which version of the chase we consider, namely the oblivious, semi-oblivious and restricted chase. The problem remains undecidable even if the database is known; this has been established in [8] for the restricted chase, and it was observed in [18] that the same proof shows undecidability also for the (semi-)oblivious chase.

The undecidability proof given in [11] constructs a sophisticated set of tgds that goes beyond existing well-behaved classes of tgds that enjoy certain syntactic properties, which in turn ensure useful model-theoretic properties. This has been already observed in [4], where it is shown that the chase termination problem is decidable if we focus on the (semi-)oblivious version of the chase, and classes of tgds based on the notion of *guardedness*. Guardedness is one of the main decidability paradigms that gives rise to robust tgd-based languages [2, 5, 6] that capture important database constraints and lightweight description logics. The key model-theoretic property of guarded-based languages, which explains their robust behaviour, is the tree-likeness of the underlying universal models [5]. On the other hand, there are interesting statements that are inherently non-tree-like, and thus not expressible via guarded-based languages. Such a statement consists of the tgds

$$\forall x \forall y (R(x, y) \rightarrow \exists z\, R(y, z) \wedge P(z)) \qquad \forall x \forall y (P(x) \wedge P(y) \rightarrow S(x, y)),$$

which compute the cartesian product of a unary relation that stores infinitely many elements.

The inability of guarded-based tgds to express non-tree-like statements like the one above, has motivated a long line of research on isolating well-behaved classes of tgds that go beyond tree-like models and guardedness. The main decidability paradigm obtained from this effort is known as *stickiness* [7]. The key idea underlying stickiness can be described as follows: variables that appear more than once in the left-hand side of a tgd, known as the body of the tgd, should be inductively propagated (or "stick") to every atom in the right-hand side of

the tgd; more details are given in Section 2. It is easy to verify that the above non-tree-like statement is trivially sticky since none of the body variables occurs more than once. The crucial question that comes up is the following: *given a sticky set Σ of tgds, is it possible to decide whether the chase terminates for every input database?*

The main goal of this work is to study the chase termination problem for sticky sets of tgds, and give a definite answer to the above fundamental question. In fact, we focus on the (semi-)oblivious versions of the chase, and we show that deciding termination for sticky sets of tgds is decidable, and provide precise complexity results: PSpace-complete in general, and NLogSpace-complete for predicates of bounded arity. Although the (semi-)oblivious versions of the chase are considered as non-standard ones, they have certain advantages that classify them as important algorithmic tools, and thus they deserve our attention. In particular, unlike the restricted chase, the application of a tgd does not require checking if the right-hand side of the tgd is already satisfied by the instance, and this guarantees technical clarity and efficiency; for a more thorough discussion on the advantages of the oblivious and semi-oblivious chase see [5, 18].

Summary of Contributions. Our results can be summarized as follows:

- In Section 4, we provide a semantic characterization of non-termination of the oblivious and semi-oblivious chase under sticky sets of tgds via the existence of path-like infinite chase derivations, which forms the basis for our decision procedure.
- By exploiting the above semantic characterization, we then provide, in Section 5, a syntactic characterization of chase termination via graph-based conditions. To this end, we extend recent syntactic characterizations from [4] of the termination of the oblivious and semi-oblivious chase under *constant-free* linear tgds (tgds with one body atom), to linear tgds with constants. The transition from constant-free tgds to tgds with constants turned out to be more challenging than expected.
- Finally, in Section 6, by exploiting the graph-based syntactic characterization from the previous section, we establish the precise complexity of our problem: PSpace-complete in general, and NLogSpace-complete for predicates of bounded arity.

2 Preliminaries

We consider the disjoint countably infinite sets $\mathbf{C}$, $\mathbf{N}$, and $\mathbf{V}$ of *constants*, *(labeled) nulls*, and (regular) *variables* (used in dependencies), respectively. A fixed lexicographic order is assumed on $(\mathbf{C} \cup \mathbf{N})$ such that every null of $\mathbf{N}$ follows all constants of $\mathbf{C}$. We refer to constants, nulls and variables as *terms*. Let $[n] = \{1, \ldots, n\}$, for any integer $n \geq 1$.

Relational Databases. A *schema* $\mathbf{S}$ is a finite set of relation symbols (or predicates) with associated arity. We write R/n to denote that R has arity $n > 0$. A position $R[i]$ in $\mathbf{S}$, where $R/n \in \mathbf{S}$ and $i \in [n]$, identifies the i-th argument of R. An *atom* over $\mathbf{S}$ is an expression of the form $R(\bar{t})$, where $R/n \in \mathbf{S}$ and $\bar{t}$ is an n-tuple of terms. We write $\mathsf{var}(\alpha)$ for the set of variables occurring in an atom α; this notation naturally extends to sets of atoms. A *fact* is an atom whose arguments consist only of constants. An *instance* over $\mathbf{S}$ is a (possibly infinite) set of atoms over $\mathbf{S}$ that contain constants and nulls, while a *database* over $\mathbf{S}$ is a finite set of facts over $\mathbf{S}$. The *active domain* of an instance I, denoted $\mathsf{dom}(I)$, is the set of all terms, i.e., constants and nulls, occurring in I.

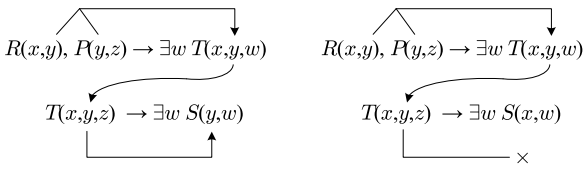
Substitutions and Homomorphisms. A *substitution* from a set T of terms to a set T' of terms is a function $h : T \to T'$ defined as follows: $\emptyset$ is a substitution (empty substitution), and if h is a substitution, then $h \cup \{t \mapsto t'\}$, where $t \in T$ and $t' \in T'$, is a substitution. The restriction of h to a subset S of T, denoted $h_{|S}$, is the substitution $\{t \mapsto h(t) \mid t \in S\}$. A *homomorphism* from a set A of atoms to a set B of atoms is a substitution h from the set of terms in A to the set of terms in B such that (i) $t \in \mathbf{C}$ implies $h(t) = t$, i.e., h is the identity on $\mathbf{C}$, and (ii) $R(t_1, \ldots, t_n) \in A$ implies $h(R(t_1, \ldots, t_n)) = R(h(t_1), \ldots, h(t_n)) \in B$.

Tuple-Generating Dependencies. A *tuple-generating dependency* σ is a sentence

$$\forall \bar{x} \forall \bar{y} \, (\phi(\bar{x}, \bar{y}) \ \rightarrow \ \exists \bar{z} \, \psi(\bar{x}, \bar{z})) \, ,$$

where $\bar{x}, \bar{y}, \bar{z}$ are tuples of variables of $\mathbf{V}$, while $\phi(\bar{x}, \bar{y})$ and $\psi(\bar{x}, \bar{z})$ are conjunctions of atoms (possibly with constants). For brevity, we write σ as $\phi(\bar{x}, \bar{y}) \rightarrow \exists \bar{z} \, \psi(\bar{x}, \bar{z})$, and use comma instead of $\wedge$ for joining atoms. We refer to $\phi(\bar{x}, \bar{y})$ and $\psi(\bar{x}, \bar{z})$ as the *body* and *head* of σ, denoted $\mathsf{body}(\sigma)$ and $\mathsf{head}(\sigma)$, respectively. The *frontier* of the tgd σ, denoted $\mathsf{fr}(\sigma)$, is the set of variables $\bar{x}$, i.e., the variables that appear both in the body and the head of σ. The *schema* of a set Σ of tgds, denoted $\mathsf{sch}(\Sigma)$, is the set of predicates in Σ. We also write $\mathsf{const}(\Sigma)$ for the set of constants occurring in Σ. An instance I satisfies a tgd σ as the one above, written $I \models \sigma$, if the following holds: whenever there exists a homomorphism h such that $h(\phi(\bar{x}, \bar{y})) \subseteq I$, then there exists $h' \supseteq h_{|\bar{x}}$ such that $h'(\psi(\bar{x}, \bar{z})) \subseteq I$. Note that, by abuse of notation, we sometimes treat a conjunction of atoms as a set of atoms. The instance I satisfies a set Σ of tgds, written $I \models \Sigma$, if $I \models \sigma$ for each $\sigma \in \Sigma$.

One of the main syntactic paradigms for tgds is stickiness [7]. The key property underlying this condition is as follows: variables that appear more than once in the body of a tgd should be inductively propagated (or "stick") to every head atom. This is graphically illustrated as



where the first set of tgds is sticky, while the second is not. The formal definition is based on an inductive procedure that marks the variables that may violate the property described above. Roughly, during the base step of this procedure, a variable that appears in the body of a tgd but not in every head atom is marked. Then, the marking is inductively propagated from head to body. Stickiness requires every marked variable to appear only once in the body of a tgd. The formal definition follows. Let Σ be a set of tgds; w.l.o.g., we assume that the tgds in Σ do not share variables. Given an atom $R(\bar{t})$ and a variable x in $\bar{t}$, $\mathsf{pos}(R(\bar{t}), x)$ is the set of positions in $R(\bar{t})$ at which x occurs. Let $\sigma \in \Sigma$ and x a variable in the body of σ. We inductively define when x *is marked in* Σ:

- If x does not occur in every atom of $\mathsf{head}(\sigma)$, then x is marked in Σ.
- Assuming that $\mathsf{head}(\sigma)$ contains an atom of the form $R(\bar{t})$ and $x \in \bar{t}$, if there exists $\sigma' \in \Sigma$ that has in its body an atom of the form $R(\bar{t}')$, and each variable in $R(\bar{t}')$ at a position of $\mathsf{pos}(R(\bar{t}), x)$ is marked in Σ, then x is marked in Σ.

The set Σ is *sticky* if there is no tgd that contains two occurrences of a variable that is marked in Σ. We denote by $\mathbb{S}$ the class of sticky finite sets of tgds. Let us clarify that we work with finite sets of tgds only. Thus, in the rest of the paper, a set of tgds is always finite.

The Tgd Chase Procedure. The tgd chase procedure (or simply chase) takes as an input a database D and a set Σ of tgds, and constructs a (possibly infinite) instance I such that $I \supseteq D$ and $I \models \Sigma$. A crucial notion is that of trigger for a set of tgds on some instance. Consider a set Σ of tgds and an instance I. A *trigger* for Σ on I is a pair (σ, h), where $\sigma \in \Sigma$ and h is a homomorphism such that $h(\mathsf{body}(\sigma)) \subseteq I$. An application of (σ, h) to I returns the instance $J = I \cup h'(\mathsf{head}(\sigma))$, where $h' \supseteq h_{|\mathsf{fr}(\sigma)}$ is such that (i) for each existentially quantified variable z of σ, $h'(z) \in \mathbf{N}$ does not occur in I and follows lexicographically all nulls in I, and (ii) for each pair (z, w) of distinct existentially quantified variables of σ, $h'(z) \neq h'(w)$. Such a trigger application is denoted as $I\langle\sigma, h\rangle J$.

The main idea of the chase is, starting from a database D, to exhaustively apply triggers for the given set Σ of tgds on the instance constructed so far. However, the choice of the next trigger to be applied is crucial since it gives rise to different variations of the chase procedure. In this work, we focus on the *oblivious* [5] and the *semi-oblivious* [12, 18] chase.

Oblivious. A finite sequence $I_0, I_1, \ldots, I_n$ of instances, where $n \geq 0$, is said to be a *terminating oblivious chase sequence* of I_0 w.r.t. a set Σ of tgds if: (i) for each $0 \leq i < n$, there exists a trigger (σ, h) for Σ on I_i such that $I_i\langle\sigma, h\rangle I_{i+1}$; (ii) for each $0 \leq i < j < n$, assuming that $I_i\langle\sigma_i, h_i\rangle I_{i+1}$ and $I_j\langle\sigma_j, h_j\rangle I_{j+1}$, $\sigma_i = \sigma_j$ implies $h_i \neq h_j$, i.e., h_i and h_j are different homomorphisms; and (iii) there is no trigger (σ, h) for Σ on I_n such that $(\sigma, h) \notin \{(\sigma_i, h_i)\}_{0 \leq i < n}$. In this case, the result of the chase is the (finite) instance I_n. An infinite sequence $I_0, I_1, \ldots$ of instances is said to be a *non-terminating oblivious chase sequence* of I_0 w.r.t. Σ if: (i) for each $i \geq 0$, there exists a trigger (σ, h) for Σ on I_i such that $I_i\langle\sigma, h\rangle I_{i+1}$; (ii) for each $i, j > 0$ such that $i \neq j$, assuming that $I_i\langle\sigma_i, h_i\rangle I_{i+1}$ and $I_j\langle\sigma_j, h_j\rangle I_{j+1}$, $\sigma_i = \sigma_j$ implies $h_i \neq h_j$; and (iii) for each $i \geq 0$, and for every trigger (σ, h) for Σ on I_i, there exists $j \geq i$ such that $I_j\langle\sigma, h\rangle I_{j+1}$; this is the *fairness condition*, and guarantees that all the triggers eventually will be applied. The result of the chase is $\bigcup_{i \geq 0} I_i$.

Semi-oblivious. This is a refined version of the oblivious chase, which avoids the application of some superfluous triggers. Roughly, given a tgd σ, for the semi-oblivious chase, two homomorphisms h and g that agree on the frontier of σ, i.e., $h_{|\mathsf{fr}(\sigma)} = g_{|\mathsf{fr}(\sigma)}$, are indistinguishable. To formalize this, we first define the binary relation $\sim_\sigma$ on the set of all possible substitutions from the terms in $\mathsf{body}(\sigma)$ to $(\mathbf{C} \cup \mathbf{N})$, denoted S_σ, as follows: $h \sim_\sigma g$ iff $h_{|\mathsf{fr}(\sigma)} = g_{|\mathsf{fr}(\sigma)}$. It is easy to verify that $\sim_\sigma$ is an equivalence relation on S_σ. A (terminating or non-terminating) oblivious chase sequence $I_0, I_1, \ldots$ is called *semi-oblivious* if the following holds: for every $i, j \geq 0$ such that $i \neq j$, assuming that $I_i\langle\sigma_i, h_i\rangle I_{i+1}$ and $I_j\langle\sigma_j, h_j\rangle I_{j+1}$, $\sigma_i = \sigma_j = \sigma$ implies $h_i \not\sim_\sigma h_j$, i.e., h_i and h_j belong to different equivalence classes.

Henceforth, we write o-chase and so-chase for oblivious and semi-oblivious chase, respectively. A useful notion that we are going to use in our proofs is the so-called chase relation [7], which essentially describes how the atoms generated during the chase depend on each other. Fix a non-terminating $\star$-chase sequence $s = (I_i)_{i \geq 0}$, where $\star \in \{\mathsf{o}, \mathsf{so}\}$, of a database D w.r.t. a set Σ of tgds, and assume that for each $i \geq 0$, $I_i\langle\sigma_i, h_i\rangle I_{i+1}$, i.e., I_{i+1} is obtained from I_i via the application of the trigger (σ_i, h_i) to I_i. The *chase relation* of s, denoted $\prec_s$, is a binary relation over $\bigcup_{i \geq 0} I_i$ such that $\alpha \prec_s \beta$ iff there exists $i \geq 0$ such that $\alpha \in h_i(\mathsf{body}(\sigma_i))$ and $\beta \in I_{i+1} \setminus I_i$.

3 Chase Termination Problem

It is well-known that due to the existentially quantified variables, a $\star$-chase sequence, where $\star \in \{\mathsf{o}, \mathsf{so}\}$, may be infinite. This is true even for very simple settings: it is easy to verify that the only $\star$-chase sequence of $D = \{R(a, b)\}$ w.r.t. the set Σ consisting of the single tgd $R(x, y) \rightarrow \exists z\, R(y, z)$ is non-terminating. The question that comes up is, given a set Σ of tgds, whether we can check that, for every database D, all or some (semi-)oblivious chase sequences of D w.r.t. Σ are terminating. Before formalizing the above problem, let us recall the following central classes of tgds:

$$\mathbb{CT}^{\star}_{\forall\forall} = \left\{ \Sigma \;\middle|\; \begin{array}{l} \text{for } \textit{every} \text{ database } D, \\ \textit{every } \star\text{-chase sequence of } D \text{ w.r.t. } \Sigma \text{ is terminating} \end{array} \right\}$$

$$\mathbb{CT}^{\star}_{\forall\exists} = \left\{ \Sigma \;\middle|\; \begin{array}{l} \text{for } \textit{every} \text{ database } D, \\ \text{there } \textit{exists} \text{ a terminating } \star\text{-chase sequence of } D \text{ w.r.t. } \Sigma \end{array} \right\}$$

The main problems tackled in this work are defined as follows, where $\mathbb{C}$ is a class of tgds:

PROBLEM :	$\mathsf{CT}^{\star}_{\forall\forall}(\mathbb{C})$
INPUT :	A set $\Sigma \in \mathbb{C}$ of tgds.
QUESTION :	Is $\Sigma \in \mathbb{CT}^{\star}_{\forall\forall}$?

PROBLEM :	$\mathsf{CT}^{\star}_{\forall\exists}(\mathbb{C})$
INPUT :	A set $\Sigma \in \mathbb{C}$ of tgds.
QUESTION :	Is $\Sigma \in \mathbb{CT}^{\star}_{\forall\exists}$?

It is well-known that $\mathbb{CT}^{\mathsf{o}}_{\forall\forall} = \mathbb{CT}^{\mathsf{o}}_{\forall\exists} \subset \mathbb{CT}^{\mathsf{so}}_{\forall\forall} = \mathbb{CT}^{\mathsf{so}}_{\forall\exists}$ [12]. This immediately implies that, after fixing the version of the chase in consideration, i.e., oblivious or semi-oblivious, the above decision problems are equivalent. Henceforth, for a class $\mathbb{C}$ of tgds, we simply refer to the problem $\mathsf{CT}^{\star}_{\forall}(\mathbb{C})$, and we write $\mathbb{CT}^{\star}_{\forall}$ for the classes $\mathbb{CT}^{\star}_{\forall\forall}$ and $\mathbb{CT}^{\star}_{\forall\exists}$, where $\star \in \{\mathsf{o}, \mathsf{so}\}$.

We know that our main problem is undecidable if we consider arbitrary tgds. In fact, assuming that $\mathbb{TGD}$ denotes the class of arbitrary tgds, we have that:

▶ **Theorem 1.** *For $\star \in \{\mathsf{o}, \mathsf{so}\}$, $\mathsf{CT}^{\star}_{\forall}(\mathbb{TGD})$ is undecidable.*

The above result has been shown in [11]. However, the employed set of tgds for showing this result is far from being sticky. This led us to ask whether $\mathsf{CT}^{\star}_{\forall}(\mathbb{S})$ is decidable. This is a non-trivial problem, and pinpointing its complexity is the main goal of this work.

Some Useful Results. Before proceeding with the complexity analysis, let us recall a couple of technical results that would allow us to significantly simplify our later analysis.

It would be useful to have a special database that gives rise to a non-terminating chase sequence if it exists. Interestingly, such a database exists, which is known as the critical database for a set of tgds [18]. Formally, given a set Σ of tgds, the *critical database* for Σ is

$$\mathsf{cr}(\Sigma) = \begin{cases} \{R(c, \ldots, c) \mid R \in \mathsf{sch}(\Sigma)\}, \text{where } c \in \mathbf{C} \text{ is a fixed constant} & \text{if } \mathsf{const}(\Sigma) = \emptyset, \\ \{R(c_1, \ldots, c_n) \mid R \in \mathsf{sch}(\Sigma) \text{ and } (c_1, \ldots, c_n) \in \mathsf{const}(\Sigma)^n\} & \text{if } \mathsf{const}(\Sigma) \neq \emptyset. \end{cases}$$

In other words, $\mathsf{cr}(\Sigma)$ consists of all the atoms that can be formed using the predicates and the constants in Σ; if Σ is constant-free, then we consider an arbitrary constant of $\mathbf{C}$. The following result from [18] shows that $\mathsf{cr}(\Sigma)$ is indeed the desired database:

▶ **Proposition 2.** *Consider a set Σ of tgds. For $\star \in \{\mathsf{o}, \mathsf{so}\}$, $\Sigma \notin \mathbb{CT}^{\star}_{\forall}$ iff there exists a non-terminating $\star$-chase sequence of $\mathsf{cr}(\Sigma)$ w.r.t. Σ.*

Even though we can focus on the critical database and check whether it gives rise to a non-terminating chase sequence s, the main difficulty is to ensure that s enjoys the fairness condition. Interestingly, as it has been recently shown in [4], we can neglect the fairness condition, which significantly simplifies the required analysis. To formalize this result, we need to recall the notion of the infinite chase derivation, which is basically a non-terminating chase sequence without the fairness condition. Fix $\star \in \{\mathsf{o}, \mathsf{so}\}$. We define $\diamond^\star_\sigma$ as $\neq$, if $\star = \mathsf{o}$, and $\not\sim_\sigma$, if $\star = \mathsf{so}$. An *infinite $\star$-chase derivation* of a database D w.r.t. a set Σ of tgds is an infinite sequence $(I_i)_{i\geq 0}$ of instances, where $I_0 = D$, such that: (i) for each $i \geq 0$, there exists a trigger (σ_i, h_i) for Σ in I_i with $I_i \langle \sigma_i, h_i \rangle I_{i+1}$, and (ii) for each $i \neq j$, $\sigma_i = \sigma_j = \sigma$ implies $h_i \diamond^\star_\sigma h_j$. The following holds:

▶ **Proposition 3.** *Consider a database D and a set Σ of tgds. For $\star \in \{\mathsf{o}, \mathsf{so}\}$, the following are equivalent:*

1. *There is a non-terminating $\star$-chase sequence of D w.r.t. Σ.*
2. *There is an infinite $\star$-chase derivation of D w.r.t. Σ.*

By combining Propositions 2 and 3, we immediately get the following useful result:

▶ **Corollary 4.** *Consider a set Σ of tgds. For $\star \in \{\mathsf{o}, \mathsf{so}\}$, $\Sigma \notin \mathbb{CT}^\star_\forall$ iff there exists an infinite $\star$-chase derivation of $\mathsf{cr}(\Sigma)$ w.r.t Σ.*

4 Semantic Characterization of Chase Non-Termination

We proceed to characterize the non-termination of the (semi-)oblivious chase under sticky sets of tgds. In particular, we show that if a sticky set Σ of tgds does not belong to $\mathbb{CT}^\star_\forall$, for $\star \in \{\mathsf{o}, \mathsf{so}\}$, then we can always isolate a *linear* infinite $\star$-chase derivation δ_ℓ of $\mathsf{cr}(\Sigma)$ w.r.t. Σ. Roughly, linearity means that there is an infinite simple path $\alpha_0, \alpha_1, \alpha_2 \ldots$ in the chase relation of δ_ℓ such that $\alpha_0 \in \mathsf{cr}(\Sigma)$ and α_i is constructed during the i-th trigger application, while all the atoms that are needed to construct this path, and are not already on the path, are atoms of $\mathsf{cr}(\Sigma)$. Notice that the chase relation of a $\star$-chase derivation is defined in the same way as the chase relation of a $\star$-chase sequence.

▶ **Definition 5.** *Consider a set Σ of tgds. For $\star \in \{\mathsf{o}, \mathsf{so}\}$, an infinite $\star$-chase derivation $\delta = (I_i)_{i\geq 0}$ of $\mathsf{cr}(\Sigma)$ w.r.t. Σ, where $I_i \langle \sigma_i, h_i \rangle I_{i+1}$ for $i \geq 0$, is called* linear *if there exists an infinite sequence of distinct atoms $(\alpha_i)_{i\geq 0}$ such that the following hold:*

- $\alpha_0 \in \mathsf{cr}(\Sigma)$.
- *For each $i \geq 0$, $\alpha_{i+1} \in I_{i+1} \setminus I_i$, and there exists $\beta \in \mathsf{body}(\sigma_i)$ such that $h_i(\beta) = \alpha_i$ and $h_i(\mathsf{body}(\sigma_i) \setminus \{\beta\}) \subseteq \mathsf{cr}(\Sigma)$.*

A simple example that illustrates the notion of linear infinite o-chase derivation follows:

▶ **Example 6.** Let Σ be the sticky set consisting of the tgd

$$\sigma = P(x, y, z), R(y, w) \rightarrow \exists v\, P(z, y, v), R(y, v).$$

Consider the infinite o-chase derivation $\delta = (I_i)_{i\geq 0}$ of $\mathsf{cr}(\Sigma)$ w.r.t. Σ, where

$$\begin{aligned}
I_0 &= \{P(c,c,c), R(c,c)\} && \langle \sigma, h_0 = \{x \mapsto c, y \mapsto c, z \mapsto c, w \mapsto c\}\rangle \\
I_1 = I_0 \cup \{P(c,c,\bot_1), R(c,\bot_1)\} && \langle \sigma, h_1 = \{x \mapsto c, y \mapsto c, z \mapsto \bot_1, w \mapsto c\}\rangle \\
I_2 = I_1 \cup \{P(\bot_1,c,\bot_2), R(c,\bot_2)\} && \langle \sigma, h_2 = \{x \mapsto \bot_1, y \mapsto c, z \mapsto \bot_2, w \mapsto c\}\rangle \\
&\vdots \\
I_{i+1} = I_i \cup \{P(\bot_i,c,\bot_{i+1}), R(c,\bot_{i+1})\} && \langle \sigma, h_{i+1} = \{x \mapsto \bot_i, y \mapsto c, z \mapsto \bot_{i+1}, w \mapsto c\}\rangle \\
&\vdots
\end{aligned}$$

Let $\alpha_0 = P(c,c,c)$, $\alpha_1 = P(c,c,\bot_1)$, and $\alpha_i = P(\bot_{i-1},c,\bot_i)$ for $i > 1$. It is easy to verify that δ is linear due to $(\alpha_i)_{i\geq 0}$. Indeed, $\alpha_0 \in \mathsf{cr}(\Sigma)$, and for every $i \geq 0$, α_i belongs to $I_{i+1} \setminus I_i$, while $h_i(P(x,y,z)) = \alpha_i$ and $h_i(R(y,w)) = R(c,c) \in \mathsf{cr}(\Sigma)$.

We are now ready to present the main characterization of non-termination of the oblivious and semi-oblivious chase under sticky sets of tgds via linear infinite $\star$-chase derivations.

▶ **Theorem 7.** *Consider a set $\Sigma \in \mathbb{S}$ of tgds. For $\star \in \{\mathsf{o}, \mathsf{so}\}$, $\Sigma \notin \mathbb{CT}^{\star}_{\forall}$ iff there exists a linear infinite $\star$-chase derivation of $\mathsf{cr}(\Sigma)$ w.r.t. Σ.*

By Corollary 4, it suffices to show the following: the existence of an infinite $\star$-chase derivation of $\mathsf{cr}(\Sigma)$ w.r.t. Σ implies the existence of a linear infinite $\star$-chase derivation of $\mathsf{cr}(\Sigma)$ w.r.t. Σ. This is a rather involved result, which is established in two main steps:

1. We show that the existence of an infinite $\star$-chase derivation of $\mathsf{cr}(\Sigma)$ w.r.t. Σ implies the existence of an infinite $\star$-chase derivation δ of $\mathsf{cr}(\Sigma)$ w.r.t. Σ such that the chase relation of δ contains a special path rooted at an atom of $\mathsf{cr}(\Sigma)$, called *continuous*. Intuitively, continuity ensures the continuous propagation of a new null on the path in question.
2. By exploiting the existence of a continuous path, we construct a linear infinite $\star$-chase derivation of $\mathsf{cr}(\Sigma)$ w.r.t. Σ. In fact, due to stickiness, we can convert an infinite suffix P of the continuous path in $\prec_\delta$, together with all the atoms that are needed to generate the atoms on P via a single trigger application, into a linear infinite $\star$-chase derivation δ_ℓ of $\mathsf{cr}(\Sigma)$ w.r.t. Σ. As we shall see, stickiness helps us to ensure that δ_ℓ is linear, while continuity allows us to show that δ_ℓ is infinite.

We proceed to give some more details for the above two steps. Although we keep the following discussion informal, we give enough evidence for the validity of Theorem 7.

4.1 Existence of a Continuous Path

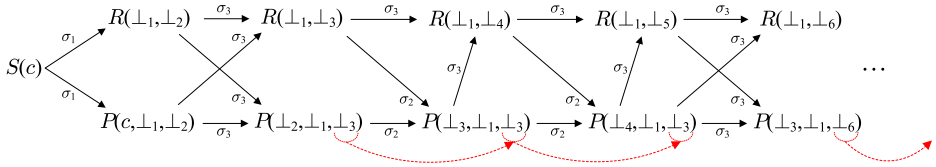
Let us first make the notion of the path in the chase relation of a derivation more precise. Given an infinite $\star$-chase derivation $\delta = (I_i)_{i\geq 0}$ of $\mathsf{cr}(\Sigma)$ w.r.t. Σ, a *finite δ-path* is a finite sequence of atoms $(\alpha_i)_{0\leq i\leq n}$ such that $\alpha_0 \in I_0$ and $\alpha_i \prec_\delta \alpha_{i+1}$. Analogously, we can define *infinite δ-paths*, which are infinite sequences of atoms rooted at an atom of I_0.

The intention underlying continuity is to ensure the continuous propagation of a new null on a path. Roughly, a δ-path $(\alpha_i)_{0\leq i\leq n}$ is continuous via a sequence of indices $(\ell_i)_{0\leq i\leq m}$, with $\ell_0 < \cdots < \ell_m$, if $\ell_0 = 1$, $\ell_m = n$, and, for each $i \in \{0,\ldots,m\}$, a new null is invented in α_{ℓ_i} that is *necessarily propagated* up to the atom $\alpha_{\ell_{i+1}}$ in case $\star = \mathsf{so}$ (resp., the atom before $\alpha_{\ell_{i+1}}$ in case $\star = \mathsf{o}$). An infinite δ-path $(\alpha_i)_{i\geq 0}$ is continuous if there exists an infinite sequence of indices $(\ell_i)_{i\geq 0}$, with $\ell_0 < \ell_1 < \cdots$, such that every finite δ-path $(\alpha_i)_{0\leq i\leq \ell_j}$, for $j \geq 0$, is continuous via $(\ell_i)_{0\leq i\leq \ell_j}$. Here is a simple example that illustrates this notion.

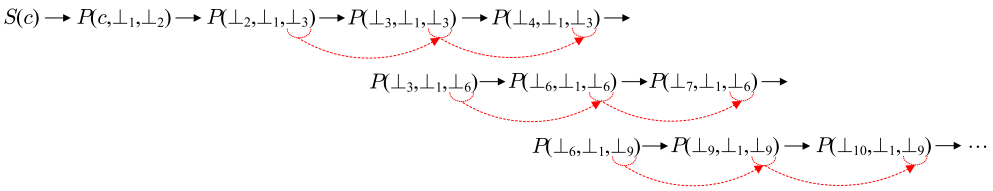
▶ **Example 8.** Consider the sticky set $\Sigma = \{\sigma_1, \sigma_2, \sigma_3\}$, where

$$\begin{aligned}
\sigma_1 &= S(x) \rightarrow \exists y \exists z\, P(x,y,z), R(y,z) \\
\sigma_2 &= P(x,y,z), R(y,w) \rightarrow P(w,y,z) \\
\sigma_3 &= P(x,y,z), R(y,w) \rightarrow \exists v\, P(z,y,v), R(y,v).
\end{aligned}$$

Notice that σ_3 is the tgd used in Example 6. It is easy to verify that there exists an infinite o-chase derivation δ of $\mathsf{cr}(\Sigma)$ w.r.t. Σ such that the following is part of $\prec_\delta$; a black edge from α to β labeled by σ means that (α, β) belongs to $\prec_\delta$ due to a trigger that involves the tgd σ:



It can be verified that the path with P-atoms in the figure is a continuous infinite δ-path. Let us explain the reason. The first atom in which a null is invented is $P(c, \bot_1, \bot_2)$, with $\bot_1, \bot_2$ being the new nulls, and continuity is satisfied since the next atom invents a null, that is, $\bot_3$. Now, since the null $\bot_3$ is propagated (this is indicated via the red dashed arrows) up to the atom before the next null generator $P(\bot_3, \bot_1, \bot_6)$, continuity is satisfied. In the rest of the path the same pattern is repeated, and thus continuity is globally satisfied. In fact, the pattern that we can extract is the following



where the continuous propagation of a new null (red arrows) can be easily observed.

We can show, via a graph-theoretic argument, that the existence of an infinite $\star$-chase derivation of $\mathsf{cr}(\Sigma)$ w.r.t. Σ implies the existence of an infinite $\star$-chase derivation of $\mathsf{cr}(\Sigma)$ w.r.t. Σ that admits a continuous infinite path. Let us briefly explain the key idea underlying this result. If we know that an infinite $\star$-chase derivation δ of $\mathsf{cr}(\Sigma)$ w.r.t. Σ exists, then we can construct an infinite $\star$-chase derivation $\delta' = (I_i)_{i \geq 0}$ of $\mathsf{cr}(\Sigma)$ w.r.t. Σ (by essentially rearranging the triggers of δ in order to obtain a derivation of a convenient form) such that the following statement holds:

there exists an infinite directed acyclic rooted graph $G = (N \cup \{\bullet\}, E, \lambda)$ of finite degree, where $\bullet$ is the root, $N \subseteq \bigcup_{i \geq 0} I_i$, every node of N is reachable from $\bullet$, and λ labels the edges of E with finite sequences of atoms from $\bigcup_{i \geq 0} I_i$, such that, for every finite path $\bullet, v_1, \ldots, v_n$, for $n \geq 1$, with $\lambda(\bullet, v_1) = (\alpha^i_j)_{0 \leq j \leq m_1 - 1}$, and $\lambda(v_{i-1}, v_i) = (\alpha^i_j)_{0 \leq j \leq m_i - 1}$,

$$\left((\alpha^i_j)_{0 \leq j \leq m_i - 1} \right)_{1 \leq i \leq n} = (\alpha_i)_{0 \leq i \leq (m_1 + \cdots + m_n) - 1}$$

is a continuous δ'-path via $(\ell_i)_{0 \leq i \leq n-1}$, where $\ell_0 = 1$ and $\ell_i = \ell_{i-1} + m_{i+1}$, for $i \in [n-1]$.

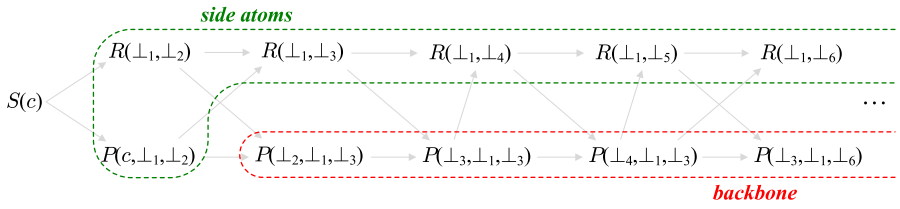
By applying König's lemma[1] on G, we get that G contains an infinite simple path $P = \bullet, v_1, v_2, \ldots$. We claim that $P' = \lambda(\bullet, v_1), \lambda(v_1, v_2), \ldots$ is a continuous δ'-path, which establishes the claim. By contradiction, assume that P' is not a continuous δ'-path. This implies that there exists a finite prefix $\bullet, v_1, \ldots, v_{n'}$ of P such that $\lambda(\bullet, v_1), \lambda(v_1, v_2), \ldots, \lambda(v_{n'-1}, v_{n'})$ is not a continuous δ'-path via $(\ell_i)_{0 \leq i \leq n'-1}$ – this is the sequence of indices used in the above statement – which contradicts the fact that the label of every finite path $\bullet, v_1, \ldots, v_n$, for $n \geq 1$, is a continuous finite δ'-path via $(\ell_i)_{0 \leq i \leq n-1}$.

4.2 From Continuous Paths to Linear Infinite Derivations

We now discuss that the existence of an infinite $\star$-chase derivation δ of $\mathsf{cr}(\Sigma)$ w.r.t. Σ such that a continuous infinite δ-path exists implies the existence of a linear infinite $\star$-chase derivation δ_ℓ of $\mathsf{cr}(\Sigma)$ w.r.t. Σ. Starting from δ, we are going to construct a sequence of instances that leads to the desired derivation δ_ℓ. The construction proceeds in three steps:

Useful part of δ. We first isolate a useful part of the $\star$-chase derivation $\delta = (I_i)_{i \geq 0}$. Recall that there exists a continuous infinite δ-path $P = (\alpha_i)_{i \geq 0}$. By stickiness, there exists $j \geq 0$ such that α_j is the last atom on P in which a term t becomes *sticky*. The latter means that the first time t participates in a join is during the trigger application that generates α_j, and thus t occurs in (or sticks to) every atom of $\{\alpha_i\}_{i \geq j}$. Let $k \geq j$ be the integer such that α_k is the first atom on P after α_j in which a new null is invented. The useful part of δ that we are going to focus on is the infinite sequence of atoms $(\alpha_i)_{i \geq k}$, which we call the *backbone*, and the atoms of $\bigcup_{i \geq 0} I_i$, which we call *side atoms*, that are needed to generate the atoms on the backbone via a single trigger application. In other words, for a backbone atom α, if α is obtained via the trigger (σ, h) for Σ on instance I_i, for some $i \geq 0$, then the atoms $h(\mathsf{body}(\sigma))$, excluding the backbone atoms, are side atoms.

▶ **Example 9.** Consider again the set $\Sigma \in \mathbb{S}$ from Example 8. As discussed above, there exists an infinite o-chase derivation δ of $\mathsf{cr}(\Sigma)$ w.r.t. Σ such that a continuous infinite δ-path exists (see the figures above). The useful part of δ is as shown below

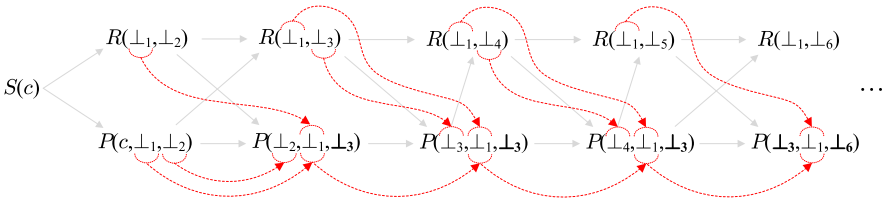


Observe that the last atom on the continuous path in which a term becomes sticky is $P(\bot_2, \bot_1, \bot_3)$; in fact, the sticky term is $\bot_1$, which is the only sticky term on the continuous path. It happened that $P(\bot_2, \bot_1, \bot_3)$ invents also a new null, that is, $\bot_3$, and therefore the suffix of the continuous path that starts at $P(\bot_2, \bot_1, \bot_3)$ is the backbone. It is now easy to verify that all the other atoms, apart from $S(c)$, indeed contribute in the generation of a backbone atom via a single trigger application.

[1] König's lemma is a well-known result from graph theory that states the following: for an infinite directed rooted graph, if every node is reachable from the root, and every node has finite out-degree, then there exists an infinite directed simple path from the root.

Renaming step. We proceed to rename some of the nulls that occur in backbone atoms or side atoms. In particular, for every null $\bot$ occurring in a side atom α, we apply the following renaming steps; fix a constant $c \in \mathsf{dom}(\mathsf{cr}(\Sigma))$: (i) every occurrence of $\bot$ in α is replaced by c, and (ii) every occurrence of $\bot$ in a backbone atom β that is propagated from α to β is replaced by c. For a backbone or side atom α, let $\rho(\alpha)$ be the atom obtained from α after globally applying the above renaming steps. We now define the sequence of instances $\delta' = (J_i)_{i \geq 0}$ as follows: $J_0 = \{\rho(\alpha) \mid \alpha \text{ is a side atom } \} \subseteq \mathsf{cr}(\Sigma)$ and $J_i = J_{i-1} \cup \{\rho(\alpha_{k+i-1})\} \cup H$, where H is the set of atoms that are generated together with α_{k+i-1} (since we can have a conjunction of atoms in the head of a tgd) after renaming the nulls that do not occur in $\rho(\alpha_{k+i-1})$ to c. Notice that H is empty in case of single-head tgds. It is crucial to observe that a new null generated in a backbone atom never participates in a join. This is because the first backbone atom α_k comes after the atom α_j, which is the last atom on P in which a term becomes sticky. This fact allows us to modify triggers from δ in order to construct, for every $i \geq 0$, a trigger (σ_i, h_i) such that $J_i \langle \sigma_i, h_i \rangle J_{i+1}$.

▶ **Example 10.** We consider again our running example. Before renaming the nulls that appear in side atoms, we first need to understand how nulls are propagated from side atoms to backbone atoms during the chase. This is depicted in the following figure



Notice that the boldfaced occurrences of the nulls $\bot_3, \bot_6, \ldots$ are not propagated from side atoms, but generated on the backbone, and thus will not be renamed. Let us recall that the existence of such nulls is guaranteed by continuity. By applying the renaming step, i.e., by replacing every null in a side atom with the constant c, and then propagating it to the backbone as indicated above, we get the sequence of instances $J_0 = \{R(c,c), P(c,c,c)\} \subseteq \mathsf{cr}(\Sigma)$, $J_1 = J_0 \cup \{P(c,c,\bot_3), R(c,\bot_3)\}$, $J_2 = J_1 \cup \{P(c,c,\bot_3)\}$, $J_3 = J_2 \cup \{P(c,c,\bot_3)\}$, $J_4 = J_3 \cup \{P(\bot_3,c,\bot_6), R(c,\bot_6)\}, \ldots$. Observe that, due to stickiness, none of the nulls $\bot_3, \bot_6, \ldots$ generated on the backbone participates in a join. This means that the renaming step preserves all the joins, and thus, by adapting triggers from δ, we can devise a valid trigger for each pair (J_i, J_{i+1}) of instances.

Pruning step. At this point, one may be tempted to think that $\delta' = (J_i)_{i \geq 0}$, with $J_i \langle \sigma_i, h_i \rangle J_{i+1}$ for $i \geq 0$, is the desired linear infinite $\star$-chase derivation of $\mathsf{cr}(\Sigma)$ w.r.t. Σ. It is easy to verify that we have the infinite sequence of atoms $(\rho(\alpha_i))_{i \geq k-1}$ such that $\rho(\alpha_{k-1}) \in \mathsf{cr}(\Sigma)$ since α_{k-1} is a side atom, and for each $i \geq k-1$, $J_{i-k+2} \supseteq J_{i-k+1} \cup \{\rho(\alpha_{i+1})\}$, and there exists $\beta \in \mathsf{body}(\sigma_i)$ such that $h_i(\beta_i) = \alpha_i$ and $h_i(\mathsf{body}(\sigma_i) \setminus \{\beta\}) \subseteq \mathsf{cr}(\Sigma)$. However, we cannot conclude yet that δ' is the desired derivation for the following two reasons:

1. triggers may repeat, i.e., we may have $i \neq j$ such that $\sigma_i = \sigma_j = \sigma$ and $h_i \diamond_\sigma^\star h_j$, where $\diamond_\sigma^\star$ is $=$ (resp., $\sim_\sigma$) if $\star = \mathsf{o}$ (resp., $\star = \mathsf{so}$), and
2. we may have $i \neq j$ such that $\rho(\alpha_i) = \rho(\alpha_j)$, i.e., the sequence of atoms $(\rho(\alpha_i))_{i \geq k-1}$ does not consist of distinct atoms.

This can be easily fixed by pruning the subderivation between the two repeated triggers or atoms. But since this pruning step may be applied infinitely many times, the question that comes up is whether the obtained $\star$-chase derivation δ'' is infinite. Interestingly, this is the case due to continuity. Since the backbone $(\alpha_i)_{i \geq k}$ is part of a continuous δ-path, we conclude that two repeated triggers or atoms are necessarily between two atoms α and β in which new nulls are invented. The fact that we have infinitely many pairs of such atoms on the backbone, we immediately conclude that after the pruning step the obtained $\star$-chase derivation is infinite. Thus, δ'' is a linear infinite $\star$-chase derivation of J_0 w.r.t. Σ. Since $J_0 \subseteq \mathsf{cr}(\Sigma)$, we can easily construct a linear infinite $\star$-chase derivation δ_ℓ of $\mathsf{cr}(\Sigma)$ w.r.t. Σ by simply adding to J_0 the set of atoms $\mathsf{cr}(\Sigma) \setminus J_0$, and the claim follows.

▶ **Example 11.** Coming back to our running example, it can be seen that the sequence of instances devised in Example 10 is not the desired linear derivation due to repeated triggers and atoms. However, after applying the pruning step, we get the sequence of instances

$$J'_0 = J_0,\ J'_1 = J_1,\ J'_2 = J'_1 \cup \{P(\bot_3, c, \bot_6), R(c, \bot_6)\},\ J'_3 = J'_2 \cup \{P(\bot_6, c, \bot_9), R(c, \bot_9)\}, \ldots.$$

Now, it is easy to verify that after adding the atom $S(c)$ in J'_0, we get (modulo null renaming) the linear infinite o-chase derivation of $\mathsf{cr}(\Sigma)$ w.r.t. Σ given in Example 6.

5 Graph-Based Characterization of Chase Termination

In this section, we characterize the termination of the (semi-)oblivious chase for sticky sets of tgds via graph-based conditions. More precisely, we show that a set $\Sigma \in \mathbb{S}$ belongs to $\mathbb{CT}^\star_\forall$ iff a linearized version of it, i.e., a set of linear tgds obtained from Σ, enjoys a condition similar to rich-acyclicity [16], if $\star = \mathsf{o}$, and weak-acyclicity [10], if $\star = \mathsf{so}$. Recall that linear tgds are tgds with only one body atom [6]; we write $\mathbb{L}$ for the class of linear tgds. The proof of the above result proceeds in two steps:

1. We first show that the given sticky set Σ of tgds can be rewritten into a set of linear tgds, while this rewriting preserves chase termination. This heavily relies on Theorem 7, which establishes that non-termination of the (semi-)oblivious chase coincides with the existence of a linear infinite chase derivation of $\mathsf{cr}(\Sigma)$ w.r.t. Σ.
2. We then extend recent characterizations from [4], which are based on extensions of rich-acyclicity and weak-acyclicity, of the termination of the (semi-)oblivious chase under constant-free linear tgds in order to deal with constants in the tgds. Although in other contexts, e.g., query answering under tgds, the transition from constant-free tgds to tgds with constants is relatively straightforward, in the context of chase termination the constants in the tgds cause additional complications that must be carefully treated.

We proceed to give more details for the above two steps.

5.1 Linearization

Before presenting the linearization procedure, we need to introduce some auxiliary notions. Given a tgd σ and an atom $\alpha \in \mathsf{body}(\sigma)$, let $V_{\alpha,\sigma} = \mathsf{var}(\mathsf{body}(\sigma) \setminus \{\alpha\})$, that is, the set of body variables of σ that do not occur only in α. Moreover, given a set Σ of tgds, a tgd $\sigma \in \Sigma$, and an atom $\alpha \in \mathsf{body}(\sigma)$, let $M^\Sigma_{\alpha,\sigma} = \{h \mid h : V_{\alpha,\sigma} \to \mathsf{dom}(\mathsf{cr}(\Sigma))\}$, i.e., the set of all possible mappings from the variables of $V_{\alpha,\sigma}$ to the constants occurring in $\mathsf{cr}(\Sigma)$.

▶ **Definition 12.** *Consider a set Σ of tgds. The* linearization *of a tgd $\sigma \in \Sigma$ (w.r.t. Σ) of the form $\phi(\bar{x}, \bar{y}) \rightarrow \exists \bar{z}\, \psi(\bar{x}, \bar{z})$, denoted $\mathsf{Lin}(\sigma)$, is the set of linear tgds*

$$\bigcup_{\alpha \in \phi(\bar{x}, \bar{y})} \bigcup_{h \in M^{\Sigma}_{\alpha,\sigma}} \{h(\alpha) \rightarrow \exists \bar{z}\, h(\psi(\bar{x}, \bar{z}))\}.$$

The linearization of Σ is defined as $\mathsf{Lin}(\Sigma) = \bigcup_{\sigma \in \Sigma} \mathsf{Lin}(\sigma)$.

The linearization procedure converts a tgd σ into a set of linear tgds by keeping only one atom α from $\mathsf{body}(\sigma)$, while the variables in $\mathsf{body}(\sigma) \setminus \{\alpha\}$ are instantiated with constants from $\mathsf{cr}(\Sigma)$ in all the possible ways. Theorem 7 allows us to show that this procedure preserves the termination of the (semi-)oblivious chase whenever the input set of tgds is sticky.

▶ **Theorem 13.** *Consider a set $\Sigma \in \mathbb{S}$ of tgds. For $\star \in \{\mathsf{o}, \mathsf{so}\}$, $\Sigma \in \mathbb{CT}^{\star}_{\forall}$ iff $\mathsf{Lin}(\Sigma) \in \mathbb{CT}^{\star}_{\forall}$.*

The ($\Rightarrow$) direction is almost immediate. Let us focus on the ($\Leftarrow$) direction. Firstly, let us clarify that it suffices to show Theorem 13 for normalized sets of tgds, i.e., tgds with only one atom in the head. This holds since $\Sigma \in \mathbb{CT}^{\star}_{\forall}$ iff $\mathsf{Norm}(\Sigma) \in \mathbb{CT}^{\star}_{\forall}$, and $\mathsf{Lin}(\Sigma) \in \mathbb{CT}^{\star}_{\forall}$ iff $\mathsf{Lin}(\mathsf{Norm}(\Sigma)) \in \mathbb{CT}^{\star}_{\forall}$, where $\mathsf{Norm}(\Sigma)$ is the normalized version of Σ obtained by applying the standard normalization procedure; see, e.g., [7]. Assume that $\Sigma \notin \mathbb{CT}^{\star}_{\forall}$. By Theorem 7, there exists a linear infinite $\star$-chase derivation $\delta = (I_i)_{i \geq 0}$ of $\mathsf{cr}(\Sigma)$ w.r.t. Σ, where $I_i \langle \sigma_i, h_i \rangle I_{i+1}$ for $i \geq 0$. In other words, there exists an infinite sequence of distinct atoms $(\alpha_i)_{i \geq 0}$ such that: (i) $\alpha_0 \in \mathsf{cr}(\Sigma)$, and (ii) for each $i \geq 0$, $\alpha_{i+1} \in I_{i+1} \setminus I_i$, and there exists $\beta_i \in \mathsf{body}(\sigma_i)$ such that $h_i(\beta_i) = \alpha_i$ and $h_i(\mathsf{body}(\sigma_i) \setminus \{\beta_i\}) \subseteq \mathsf{cr}(\Sigma)$. Let Σ' be the set of linear tgds $\{g_i(\beta_i) \rightarrow g_i(\mathsf{head}(\sigma_i))\}_{i \geq 0}$, where g_i is the restriction of h_i over V_{β_i, σ_i}. Since $\Sigma' \subseteq \mathsf{Lin}(\Sigma)$, it suffices to show that $\Sigma' \notin \mathbb{CT}^{\star}_{\forall}$. To this end, we are going to construct an infinite $\star$-chase derivation δ' of $\mathsf{cr}(\Sigma') \subseteq \mathsf{cr}(\Sigma)$ w.r.t. Σ'. The derivation δ' is obtained from δ by replacing each trigger (σ_i, h_i), for $i \geq 0$, with (σ'_i, h'_i), where σ'_i is the linear tgd $g_i(\beta_i) \rightarrow g_i(\mathsf{head}(\sigma_i))$ that belongs to Σ', while h'_i is the restriction of h_i to the terms occurring in $\mathsf{body}(\sigma'_i)$. It remains to show that δ' is indeed a valid infinite $\star$-chase derivation of $\mathsf{cr}(\Sigma')$ w.r.t. Σ', which boils down to showing that, for every $i \neq j \geq 0$, $\sigma'_i = \sigma'_j = \sigma$ implies $h'_i \diamond^{\star}_{\sigma} h'_j$, where $\diamond^{\mathsf{o}}_{\sigma}$ is $\neq$, and $\diamond^{\mathsf{so}}_{\sigma}$ is $\not\sim_{\sigma}$. Towards a contradiction, assume that there exists $i \neq j$ such that $\sigma'_i = \sigma'_j = \sigma$ but $h'_i \diamond^{\star}_{\sigma} h'_j$ does not hold. This implies that $\alpha_i = \alpha_j$ since we consider single-head tgds. But this contradicts the fact that $\alpha_i \neq \alpha_j$.

5.2 Acyclicity Conditions

We proceed to extend the characterizations of the termination of the (semi-)oblivious chase under constant-free linear tgds established in [4]. The goal is to show that, given a set of tgds $\Sigma \in \mathbb{L}$, which may contain constants, $\Sigma \in \mathbb{CT}^{\mathsf{o}}_{\forall}$ iff Σ is critically-richly-acyclic, and $\Sigma \in \mathbb{CT}^{\mathsf{so}}_{\forall}$ iff Σ critically-weakly-acyclic, where critical-rich- and critical-weak-acyclicity are appropriate extensions of rich- and weak-acyclicity proposed in [4]. These notions rely on the dependency graph of a set of tgds, which we now recall. We assume a fixed order on the head-atoms of tgds. For a tgd σ with $\mathsf{head}(\sigma) = \alpha_1, \ldots, \alpha_k$, we write (σ, i) for the *single-head* tgd, i.e., the tgd with only one atom in its head, obtained from σ by keeping only the atom α_i, and the existentially quantified variables in α_i. Recall that $\mathsf{pos}(\alpha, x)$ is the set of positions in α at which x occurs. Analogously, we write $\mathsf{pos}(\mathsf{body}(\sigma), x)$ for the set of positions at which the variable x occurs in the body of σ. We also write $\mathsf{pos}(\mathsf{sch}(\Sigma))$ for the set of positions of $\mathsf{sch}(\Sigma)$, i.e., the set $\{R[i] \mid R/n \in \mathsf{sch}(\Sigma) \text{ and } i \in \{1, \ldots, n\}\}$.

▶ **Definition 14.** *The* dependency graph *of a set Σ of tgds is a labeled directed multigraph $\mathsf{dg}(\Sigma) = (N, E, \lambda)$, where $N = \mathsf{pos}(\mathsf{sch}(\Sigma))$, $\lambda : E \to \Sigma \times \mathbb{N}$, and E contains only the following edges. For each $\sigma \in \Sigma$ with $\mathsf{head}(\sigma) = \alpha_1, \ldots, \alpha_k$, for each $x \in \mathsf{fr}(\sigma)$, and for each $\pi \in \mathsf{pos}(\mathsf{body}(\sigma), x)$:*

- *For each $i \in [k]$, and for each $\pi' \in \mathsf{pos}(\alpha_i, x)$, there is a* normal *edge $e = (\pi, \pi') \in E$ with $\lambda(e) = (\sigma, i)$.*
- *For each existentially quantified variable z in σ, for each $i \in [k]$, and for each $\pi' \in \mathsf{pos}(\alpha_i, z)$, there is a* special *edge $e = (\pi, \pi') \in E$ with $\lambda(e) = (\sigma, i)$.*

A normal edge (π, π') keeps track of the fact that a term may propagate from π to π' during the chase. Moreover, a special edge (π, π'') keeps track of the fact that the propagation of a value from π to π' also creates a null at position π''. As we shall see, the dependency graph is appropriate when we consider the semi-oblivious chase. For the oblivious chase, we need an extended version of it. The *extended dependency graph* of Σ, denoted $\mathsf{edg}(\Sigma)$, is obtained from $\mathsf{dg}(\Sigma)$ by simply adding special labeled edges from the positions where non-frontier variables occur to the positions where existentially quantified variables occur.

Two well-known classes of tgds, introduced in the context of data exchange, that guarantee the termination of the oblivious and semi-oblivious chase are rich-acyclicity and weak-acyclicity, respectively. A set Σ is richly-acyclic (resp., weakly-acyclic) if there is no cycle in $\mathsf{edg}(\Sigma)$ (resp., $\mathsf{dg}(\Sigma)$) that contains a special edge. It would be very useful if, whenever we focus on linear tgds, rich- and weak-acyclicity are also necessary conditions for the termination of the oblivious and semi-oblivious chase, respectively. Unfortunately, this is not the case. A simple counterexample follows:

▶ **Example 15.** Consider the set Σ of linear tgds consisting of $R(x, x) \to \exists z\, R(z, x)$. In $\mathsf{dg}(\Sigma) = \mathsf{edg}(\Sigma)$ there is a cycle that contains a special edge. However, for $\star \in \{\mathsf{o}, \mathsf{so}\}$, there is only one $\star$-chase sequence of $\mathsf{cr}(\Sigma)$ w.r.t. Σ that is terminating; thus, $\Sigma \in \mathbb{CT}^{\star}_{\forall}$.

As it has been shown in [4], there is an extension of rich- and weak-acyclicity, called critical-rich- and critical-weak-acyclicity, that whenever we focus on linear tgds provides a necessary and sufficient condition for the termination of the oblivious and semi-oblivious chase, respectively. However, the analysis performed in [4] considers only tgds without constants, while after the linearization of a sticky set Σ of tgds, even if Σ is constant-free, the obtained set $\mathsf{Lin}(\Sigma)$ contains at least one constant. Thus, in order to be able to apply critical-rich- and critical-weak-acyclicity on $\mathsf{Lin}(\Sigma)$, we first need to appropriately extend these notions to linear tgds with constants.

A crucial notion underlying critical-rich- and critical-weak-acyclicity is the notion of compatibility among two single-head linear tgds. Intuitively, if a single-head linear tgd σ_1 is compatible with a single-head linear tgd σ_2, then the atom obtained during the chase by applying σ_1 may trigger σ_2. It is clear that the presence of constants in the tgds affects the way that we define compatibility. We assume the reader is familiar with the notion of unification. Given two atoms α, β, we write $\mathsf{mgu}(\alpha, \beta)$ for their most general unifier. For brevity, we write Π^{σ}_{t} for the set of positions $\mathsf{pos}(\mathsf{body}(\sigma), t)$, i.e., the set of positions at which the term t occurs in the body of σ. We also write $\mathsf{term}(\alpha, \Pi)$, where α is an atom, and Π a set of positions, for the set of terms occurring in α at positions of Π.

▶ **Definition 16.** *Consider two single-head linear tgds σ_1 and σ_2. We say that σ_1 is* compatible *with σ_2 if the following hold:*

1. $\mathsf{head}(\sigma_1)$ *and* $\mathsf{body}(\sigma_2)$ *unify.*
2. *For each $x \in \mathsf{var}(\mathsf{body}(\sigma_2))$, either $\mathsf{term}(\mathsf{head}(\sigma_1), \Pi^{\sigma_2}_{x}) = \{z\}$ for some existentially quantified variable z in σ_1, or $\mathsf{term}(\mathsf{head}(\sigma_1), \Pi^{\sigma_2}_{x}) \subseteq \mathsf{fr}(\sigma_1) \cup \{c\}$ for some constant c.*
3. *For each $c \in \mathsf{const}(\mathsf{body}(\sigma_2))$, $\mathsf{term}(\mathsf{head}(\sigma_1), \Pi^{\sigma_2}_{c}) \subseteq \mathsf{fr}(\sigma_1) \cup \{c\}$.*

Having the notion of compatibility among two single-head linear tgds in place, we can recall the resolvent of a sequence $\sigma_1, \ldots, \sigma_n$ of single-head linear tgds, which is in turn a single-head tgd. Roughly, such a resolvent mimics the behavior of the sequence $\sigma_1, \ldots, \sigma_n$ during the chase. Notice that the existence of such a resolvent is not guaranteed, but if it exists, this implies that we may have a sequence of trigger applications that involve the tgds $\sigma_1, \ldots, \sigma_n$ in this order. In such a case, we call the sequence $\sigma_1, \ldots, \sigma_n$ active.

▶ **Definition 17.** *The* resolvent *of a sequence $\sigma_1, \ldots, \sigma_n$ of single-head linear tgds, denoted $[\sigma_1, \ldots, \sigma_n]$, is inductively defined as follows; for brevity, we write ρ for $[\sigma_1, \ldots, \sigma_{n-1}]$:*

1. $[\sigma_1] = \sigma_1$*;*
2. $[\sigma_1, \ldots, \sigma_n] = \gamma(\mathsf{body}(\rho)) \to \gamma(\mathsf{head}(\sigma_n))$*, where* $\gamma = \mathsf{mgu}(\mathsf{head}(\rho), \mathsf{body}(\sigma_n))$*, if* $\rho \neq \Diamond$ *and ρ is compatible with σ_n; otherwise,* $[\sigma_1, \ldots, \sigma_n] = \Diamond$*.*

The sequence $\sigma_1, \ldots, \sigma_n$ is called active *if* $[\sigma_1, \ldots, \sigma_n] \neq \Diamond$*.*

At this point, one may think that the right extension of rich- and weak-acyclicity, which will provide a necessary condition for the termination of the oblivious and semi-oblivious chase under linear tgds, is to allow cycles with special edges in the underlying dependency graph as long as the corresponding sequence of single-head tgds, which can be extracted from the edge labels, is not active. This is not enough. If a cycle with a special edge is labeled with an active sequence, then we can conclude that it will be traversed at least once during the chase. However, it is not guaranteed that it will be traversed infinitely many times.

▶ **Example 18.** Consider the set Σ of linear tgds consisting of

$$\sigma_1 = R(x, y, z) \;\to\; P(x, y, z) \qquad \sigma_2 = P(x, y, x) \;\to\; \exists z\, R(y, z, x).$$

In $\mathsf{dg}(\Sigma) = \mathsf{edg}(\Sigma)$ there is an active cycle that contains a special edge; e.g., $C = R[2], P[2], R[2]$, which corresponds to the sequence of tgds σ_1, σ_2. It is easy to see that $[\sigma_1, \sigma_2] \neq \Diamond$, and thus C is active. Despite the existence of an active cycle that contains a special edge, we can show that $\Sigma \in \mathbb{CT}^{\star}_{\forall}$, where $\star \in \{\mathsf{o}, \mathsf{so}\}$.

A cycle that is labeled with an active sequence $\sigma_1, \ldots, \sigma_n$, and contains a special edge, will be certainly traversed infinitely many times if the resolvent of the sequence $\rho, \ldots, \rho$ of length k, where $\rho = [\sigma_1, \ldots, \sigma_n]$, exists, for every $k > 0$. Interestingly, for ensuring the latter condition, it suffices to consider sequences of length at most $(\omega + 1)$, where ω is the arity of the predicate of $\mathsf{body}(\sigma_1)$. This brings us to critical sequences. For brevity, we write σ^k for the sequence $\sigma, \ldots, \sigma$ of length k.

▶ **Definition 19.** *A sequence $\sigma_1, \ldots, \sigma_n$ of single-head linear tgds is* critical *if $\sigma_1, \ldots, \sigma_n$ is active, and $[\sigma_1, \ldots, \sigma_n]^{\omega+1}$ is active, where ω is the arity of the predicate of* $\mathsf{body}(\sigma_1)$*.*

We can now recall critical-rich- and critical-weak-acyclicity. They are essentially rich- and weak-acyclicity, with the difference that a cycle in the underlying graph is "dangerous", not only if it contains a special edge, but if it is also labeled with a critical sequence.

▶ **Definition 20.** *Consider a set $\Sigma \in \mathbb{L}$ of tgds, and let $G = (N, E, \lambda)$ be either* $\mathsf{edg}(\Sigma)$ *or* $\mathsf{dg}(\Sigma)$*. A cycle $v_0, v_1, \ldots, v_n, v_0$ in G is* critical *if $\lambda(v_0, v_1), \lambda(v_1, v_2), \ldots, \lambda(v_n, v_0)$ is critical. We say that Σ is* critically-richly-acyclic *(resp.,* critically-weakly-acyclic*), if no critical cycle in* $\mathsf{edg}(\Sigma)$ *(resp.,* $\mathsf{dg}(\Sigma)$*) contains a special edge.*

The desired result follows:

▶ **Theorem 21.** *Consider a set $\Sigma \in \mathbb{L}$ of tgds. The following hold:*

- $\Sigma \in \mathbb{CT}^{\mathsf{o}}_{\forall}$ *iff* Σ *is critically-richly-acyclic.*
- $\Sigma \in \mathbb{CT}^{\mathsf{so}}_{\forall}$ *iff* Σ *is critically-weakly-acyclic.*

The "if" directions of the above result are shown by giving proofs similar to the ones given in [16] and [10] for showing that rich-acyclicity and weak-acyclicity guarantees the termination of the oblivious and restricted chase, respectively. The interesting direction is the "only if" direction. By Corollary 4, it suffices to show that if Σ is not critically-richly-acyclic (resp., critically-weakly-acyclic), then there exists an infinite o-chase (resp., so-chase) derivation of $\mathsf{cr}(\Sigma)$ w.r.t. Σ. This is a non-trivial result that requires a couple of auxiliary lemmas.

The equality type of an atom is a set of equalities among positions, as well as among positions and constants, that describes its shape. Formally, for an atom $\alpha = R(t_1, \ldots, t_n)$, the *equality type* of α is $\mathsf{eqtype}(\alpha) = \{R[i] = R[j] \mid t_i = t_j\} \cup \{R[i] = c \mid c \in \mathbf{C} \text{ and } t_i = c\}$. For a linear tgd σ, we write $\mathsf{eqtype}(\sigma)$ for the equality type of the atom $\mathsf{body}(\sigma)$. The next result establishes a useful connection between active sequences and equality types:

▶ **Lemma 22.** *Consider a single-head linear tgd σ such that σ^i is active, for some $i > 1$, and $\mathsf{eqtype}([\sigma^{i-1}]) = \mathsf{eqtype}([\sigma^i])$. Then, σ^{i+1} is active, and $\mathsf{eqtype}([\sigma^i]) = \mathsf{eqtype}([\sigma^{i+1}])$.*

Despite the fact that the above lemma has been already shown in [4] for constant-free tgds, it turned out that the proof from [4] cannot be easily extended to tgds with constants. Thus, we had to devise a completely new proof that exploits further properties of the resolvent of a sequence $\sigma, \ldots, \sigma$. In fact, we show via an inductive argument that $[\sigma^i] = [[\sigma^{i-1}], \sigma]$ is the same (modulo variable renaming) as $[\sigma, [\sigma^{i-1}]]$, which in turn allows us to easily establish Lemma 22. Having the connection between active sequences and equality types provided by Lemma 22, we show the next lemma, which states that critical cycles can be traversed infinitely many times during the chase.

▶ **Lemma 23.** *Consider a critical sequence $\sigma_1, \ldots, \sigma_n$ of single-head linear tgds. For every $k > 0$, $[\sigma_1, \ldots, \sigma_n]^k$ is active.*

As for Lemma 22, even though the above result has been shown in [4] for constant-free tgds, we had to provide a new proof in order to deal with the constants in the tgds. Let us briefly explain how Lemma 23 is shown. For brevity, let $\rho = [\sigma_1, \ldots, \sigma_n]$. Since $\sigma_1, \ldots, \sigma_n$ is critical, by definition we get that $\rho^{\omega+1}$ is active, where ω is the arity of the predicate of $\mathsf{body}(\sigma_1)$. The crucial step is to also show that $\mathsf{eqtype}([\rho^\omega]) = \mathsf{eqtype}([\rho^{\omega+1}])$. Then, by iteratively applying Lemma 22, we obtain that ρ^k is active for every $k > \omega + 1$. Since $\rho^{\omega+1}$ is active, we can conclude that ρ^k is active for every $1 \leq k \leq \omega + 1$, and Lemma 23 follows.

We can show via an inductive argument that an active sequence $\sigma_1, \ldots, \sigma_n$ of single-head linear tgds mimics the sequence of trigger applications that involve the tgds $\sigma_1, \ldots, \sigma_n$ (in this order), starting from an atom in the critical instance; in particular, the ground version of $\mathsf{body}([\rho^{\omega+1}])$. This fact and Lemma 23 allow us to show that a critical cycle of minimal length in the (extended) dependency graph that contains a special edge, gives rise to an infinite o-chase (so-chase) derivation of $\mathsf{cr}(\Sigma)$ w.r.t. Σ, and Theorem 21 follows.

It is now easy to see that Theorems 13 and 21 establish the main result of this section:

▶ **Corollary 24.** *Consider a set $\Sigma \in \mathbb{S}$ of tgds. The following hold:*

- $\Sigma \in \mathbb{CT}^{\mathsf{o}}_{\forall}$ *iff* $\mathsf{Lin}(\Sigma)$ *is critically-richly-acyclic.*
- $\Sigma \in \mathbb{CT}^{\mathsf{so}}_{\forall}$ *iff* $\mathsf{Lin}(\Sigma)$ *is critically-weakly-acyclic.*

6 Complexity of Chase Termination

In this final section, we pinpoint the complexity of the $\star$-chase termination problem under sticky sets of tgds. In particular, we establish the following complexity result:

▶ **Theorem 25.** *For* $\star \in \{\mathsf{o}, \mathsf{so}\}$, $\mathsf{CT}^{\star}_{\forall}(\mathbb{S})$ *is* PSPACE*-complete, and* NLOGSPACE*-complete for predicates of bounded arity. The lower bounds hold even for tgds without constants.*

Upper Bounds. The problem $\mathsf{CT}^{\star}_{\forall}$ under constant-free linear tgds is PSPACE-complete, in general, and NLOGSPACE-complete for predicates of bounded arity [4]. However, despite the fact that, by Corollary 24, we can reduce $\mathsf{CT}^{\star}_{\forall}(\mathbb{S})$ to $\mathsf{CT}^{\star}_{\forall}(\mathbb{L})$, we cannot directly exploit the complexity results from [4] for two reasons: (i) the linearized version of Σ contains at least one constant, while the results from [4] apply only to constant-free tgds, and (ii) the linearization procedure takes exponential time, in general, and polynomial time in the case of bounded-arity predicates; thus, we cannot explicitly compute the set $\mathsf{Lin}(\Sigma)$, and then check for critical-rich- and critical-weak-acyclicity. Therefore, a more refined procedure is needed.

We focus on the complement of our problem, i.e., given a set $\Sigma \in \mathbb{S}$ of tgds, we want to check whether $\Sigma \notin \mathbb{CT}^{\star}_{\forall}$. By Corollary 24, it suffices to show that $\mathsf{Lin}(\Sigma)$ is not critically-richly-acyclic, if $\star = \mathsf{o}$, and not critically-weakly-acyclic, if $\star = \mathsf{so}$. The latter problems can be seen as a generalization of the standard graph reachability problem. Indeed, we need to check whether there exists a node v in the (extended) dependency graph of $\mathsf{Lin}(\Sigma)$ that is reachable from itself via a critical cycle that contains a special edge. However, as discussed above, we cannot explicitly construct $\mathsf{Lin}(\Sigma)$ and its (extended) dependency graph G. Instead, the above reachability check should be performed on a compact representation of G, which is the set Σ itself. We show that this check can be performed via a non-deterministic procedure that uses $O(\omega \log(\omega \cdot |\mathsf{sch}(\Sigma)|) + \omega \log(\omega \cdot m \cdot |\Sigma|))$ space, where ω is the maximum arity over all predicates in Σ, and m is the maximum number of atoms occurring in a tgd of Σ.

Lower Bounds. The PSPACE-hardness is shown by providing a polynomial time reduction from the acceptance problem of a deterministic polynomial space Turing machine M. Such a reduction can be easily devised if we are allowed to join a variable in the body of a tgd and then lose it, or if we can use constants in the body of a tgd. In this case, a configuration of M can be straightforwardly encoded in a single predicate *Config* of polynomial arity. However, if we want the set of tgds to be sticky and constant-free, then we need a more clever encoding for a configuration of M, which increases the arity of *Config*, but only polynomially.

The NLOGSPACE-hardness is inherited from [4], where it is shown that $\mathsf{CT}^{\star}_{\forall}(\mathbb{L})$ is NLOGSPACE-hard, even for tgds that are constant-free, each body variable occurs only once (i.e., stickiness is trivially satisfied), and only unary and binary predicates are used.

7 Conclusions

We have shown that the uniform (semi-)oblivious chase termination problem for sticky sets of tgds is decidable, and obtained precise complexity results. This is done by first characterizing the termination of the (semi-)oblivious chase for sticky sets of tgds via graph-based conditions that are of independent interest. In particular, to check whether the oblivious (resp., semi-oblivious) chase terminates for a sticky set Σ of tgds, we simply need to linearize it, i.e., convert it, via an easy procedure, into a set $\mathsf{Lin}(\Sigma)$ of linear tgds, and then check whether $\mathsf{Lin}(\Sigma)$ enjoys an acyclicity condition in the spirit of rich-acyclicity (resp., weak-acyclicity). The next natural step is to concentrate on the restricted (a.k.a. standard) version of the chase, which makes the problem even more challenging due to its non-deterministic behaviour that cannot be captured via static graph-based conditions.

References

1. Alfred V. Aho, Yehoshua Sagiv, and Jeffrey D. Ullman. Efficient Optimization of a Class of Relational Expressions. *ACM Trans. Database Syst.*, 4(4):435–454, 1979.
2. Jean-François Baget, Michel Leclère, Marie-Laure Mugnier, and Eric Salvat. On rules with existential variables: Walking the decidability line. *Artif. Intell.*, 175(9-10):1620–1654, 2011.
3. Catriel Beeri and Moshe Y. Vardi. A Proof Procedure for Data Dependencies. *J. ACM*, 31(4):718–741, 1984.
4. Marco Calautti, Georg Gottlob, and Andreas Pieris. Chase Termination for Guarded Existential Rules. In *PODS*, pages 91–103, 2015.
5. Andrea Calì, Georg Gottlob, and Michael Kifer. Taming the Infinite Chase: Query Answering under Expressive Relational Constraints. *J. Artif. Intell. Res.*, 48:115–174, 2013.
6. Andrea Calì, Georg Gottlob, and Thomas Lukasiewicz. A general Datalog-based framework for tractable query answering over ontologies. *J. Web Sem.*, 14:57–83, 2012.
7. Andrea Calì, Georg Gottlob, and Andreas Pieris. Towards more expressive ontology languages: The query answering problem. *Artif. Intell.*, 193:87–128, 2012.
8. Alin Deutsch, Alan Nash, and Jeff B. Remmel. The Chase Revisisted. In *PODS*, pages 149–158, 2008.
9. Alin Deutsch and Val Tannen. Reformulation of XML Queries and Constraints. In *ICDT*, pages 225–241, 2003.
10. Ronald Fagin, Phokion G. Kolaitis, Renée J. Miller, and Lucian Popa. Data exchange: semantics and query answering. *Theor. Comput. Sci.*, 336(1):89–124, 2005.
11. Tomasz Gogacz and Jerzy Marcinkowski. All-Instances Termination of Chase is Undecidable. In *ICALP*, pages 293–304, 2014.
12. Gösta Grahne and Adrian Onet. Anatomy of the Chase. *Fundam. Inform.*, 157(3):221–270, 2018.
13. Bernardo Cuenca Grau, Ian Horrocks, Markus Krötzsch, Clemens Kupke, Despoina Magka, Boris Motik, and Zhe Wang. Acyclicity Notions for Existential Rules and Their Application to Query Answering in Ontologies. *J. Artif. Intell. Res.*, 47:741–808, 2013.
14. Sergio Greco, Cristian Molinaro, and Francesca Spezzano. *Incomplete Data and Data Dependencies in Relational Databases*. Morgan & Claypool Publishers, 2012.
15. Sergio Greco, Francesca Spezzano, and Irina Trubitsyna. Stratification Criteria and Rewriting Techniques for Checking Chase Termination. *PVLDB*, 4(11):1158–1168, 2011.
16. André Hernich and Nicole Schweikardt. CWA-solutions for data exchange settings with target dependencies. In *PODS*, pages 113–122, 2007.
17. David Maier, Alberto O. Mendelzon, and Yehoshua Sagiv. Testing Implications of Data Dependencies. *ACM Trans. Database Syst.*, 4(4):455–469, 1979.
18. Bruno Marnette. Generalized schema-mappings: from termination to tractability. In *PODS*, pages 13–22, 2009.
19. Michael Meier, Michael Schmidt, and Georg Lausen. On Chase Termination Beyond Stratification. *PVLDB*, 2(1):970–981, 2009.